caGrid Developer's Face to Face

Wednesday, April 23, 2008
8:56 AM

Subject	caGrid Developer's Face to Face
Attendees	Shannon Hastings; Stephen A. Langella; David Ervin; Ravi Madduri; Manav Kher; joshua; Modi, Kunal (NIH/NCI) [C]

Link to Outlook item

- **Release Debrief**
  - **Good**
    - Deployment process/document
    - Training deployment using individual service addresses
    - Release process (portal separate)
    - Good speed and amount of features addressed
    - Portal: documenting QA fixes and how to test
    - Getting SDK 4 support out, and testing for that is much more complete
    - Overall improvements in systematic testing
    - Communication between caBIG teams (SDK, caDSR, CCTS, etc)
    - Portal release separate and before caGrid
    - WebSSO/CDS incubation process (close interaction with user)
    - Early access from WebSSO/CDS/Transfer
    - Movement to Ivy for better dependency management
  - **Bad**
    - Issues not identified which would have been if services were upgraded
    - Training deployment using VMs (timing issues)
    - Not enough QA (need to articulate scenarios)
    - Installer QA is tedious and time consuming
    - Need monitoring service of production grid
    - Installer maintenance; need a dedicated resource or component owner (seen as user frontend to caGrid)
    - Portal: not enough QA
    - Management communication (multiple managers/manager transitions); seems resolved now
    - We need more resources :)
    - Large size

- **Testing Discussion**
  - Overview of plans for restructuring of test locations (in repository)
    - Unit, system, integration, deployment, scenario
  - Plans to work with QA team
    - Scenario testing (mostly involving GUIs)
    - On multiple platforms
    - Duration: Milestone/feature based, prior to release
    - **Action Item**: Create wiki page to start to capture scenarios
      - http://www.cagrid.org/wiki/CaGrid:TestingScenarios

- **Ivy Discussion**
  - Is there a way to ignore a set of dependencies? (can we generate an "ignore list" for globus location)
  - Model all of Globus's dependencies
    - Service just needs compile time dependencies
    - Client needs runtime dependencies
  - How to handle conflicts then (introducing dependencies on jars you don't need, but are in Globus)?
  - **Action Item:** Scott look into whether there is a way to handle things as provided (such as Maven system/provided scope)

- **Future Planning**
  - **Incubation**
    - Current Status
      - Incubation project on NCI gforge
      - Similar project layout to caGrid
        - Leverages Ivy to depend on 1.2 caGrid release
    - Going forward need to develop: Policies, Management, Lifecycle (retirement, movement into caGrid, other?)

- **ONIX collaboration**
  - Has similar requirements about metadata authenticity (want to make third party assertions about services; XML signature)
  - Relevancy for framework for making assertions about "things" (metadata in portal, schemas in gme, etc); ability to reason over

- **Workflow**
  - **Taverna Integration**
    - Taverna2
    - Metadata-based search
    - Remote Execution Service
      - Claim is its not sufficient and needs work?
    - Dates:
      - June annual meeting (stable release, not using remote invocation)
      - caGrid core release
    - Release packaging
      - Should be standalone distribution (with all our plugins installed), with documentation

caGrid Developer Face to Face Agenda

Rockville, MD
April 23rd-25th 2008

Overview

The developer face to face will be a 2.5 day meeting, attended by all caGrid developers, and focus on release planning and new feature design between all the projects in the caGrid suite. The first two full days will be internal to the caGrid team, and the final half day will invite additional members from caCORE products to focus on inter-project dependency design and planning. These will take the form of breakout sessions.

Day One: Release Debrief and Future Planning/ caGrid Suite Planning
Day Two: Design Sessions
Day Three: Cross Product Integration

Detail Agenda

Day One (9am – 5pm, Room # 5034)

Morning Session (overview)

- caGrid 1.2 Release debrief
  - Lessons learned, issues to address in next release
- Testing Planning
  - Discuss restructuring of test locations (in repository)
    - Unit, system, integration, deployment, scenario
  - Plans to work with QA team
- Future planning
  - Next Release Scoping
    - Review of previously identified uncompleted tasks
    - Review of new priorities
  - Incubation Process
    - Policies, Processes, etc
  - Planning for Milestone releases
  - Planning for long term
    - Outcomes of caGrid Roadmap meeting
    - Long term backwards compatibility issues
    - caGrid 2.0, new Globus versions (protocol incompatible)
- Build Process Review/Planning
  - review of 1.2 Ivy/build system review
  - Discussion of how to handle GLOBUS_LOCATION
- 3-4pm: Metrics discussion

Afternoon Session (design/caGrid suite)

- Workflow
  - Security integration into Taverna
  - Scope plan review
  - Packaging/release structure
- Portal
  - ONIX collaboration
  - Potential for domain resuse
- Logistic planning of joint releases (caGrid Suite)
  - portal needs/use cases
  - workflow needs/use cases

Day Two (9am – 5pm, Room #5001)

- Gold Enforcement
  - Metadata authenticity
  - Secure Index Service
- GME enhancements design review
  - Plans for new service rollout
  - How to leverage GME/caDSR binding
- Workflow service
  - Plans
  - Notification support
  - Delegation support
  - Taverna/workflow client integration (with service)
- Future planning for gravi/TG integration
- Installer review/potential feature planning
  - Installer features
  - Unattended installation
  - Can it leverage ivy?
  - "simple installation"
- CCTS needs
  - Transactions
  - Reliable messaging
- Security Design
  - Secure Authentication Service/Dorian Discovery
  - Auditing

Day Three (9am – 1pm, Room #5012)

- caCORE SDK Integration
  - CQL2
  - Data Service Integration testing plans (HEAD/Release)

and examples that can be run out of the box
- **Workflow Service**
  - Notification Support
  - Delegation Support
  - Scufl adapter
  - What will be the long term plan for workflow support?
    - Phase one language out or the other?
    - Support multiple?
    - Con against multiple is a workflow description is valuable knowledge capture, and supporting multiple languages dilutes the value
    - Resources are needed for either:
      - Taverna/Scufl needs work to support security (infrastructure) and security configuration (infrastructure/tooling) and rich metadata (client)
      - BPEL needs work to support client tooling (client metadata) and security configuration (infrastructure/tooling)
  - Big missing features:
    - Workflow construction using semantic and structural metadata
    - Security annotation and enforcement (ability to say execute this operation as me or anonymously)

- **Metadata Authenticity**
  - Need to identify the use cases we need to support
  - "best way" is to have service use service key to sign service metadata, then have an authoritative key sign that
    - Issues about where validation of signatures are done (in client, in index service (at publish time or at query time))
  - May be able to have an authoritative source, such as caDSR making assertions (could just use hashes of the metadata and service URL as an approximation) for things like Gold compliance
  - Could leverage something like an "assertion repository" (see above) to make these statements; this would allow us to start with the hash approach then add the key approach if needed

- **Backwards compatibility**
  - Aspects of compatibility
    - API level
    - Grid/Spec level
    - Globus version (4.2 is coupled to spec level)
    - Grid deployment
  - Movement to GT4.2:
    - Implies a change in specifications (wsrf, addressing, ws-rp, ws-n)
      - Breaks backwards compatibility in the grid
      - Some API level changes (e.g. some operation name changes in WS-N)
      - Large effort to support in Introduce (service upgrade and new service generation)
  - We need a plan to deprecate things (e.g probably can't support all of 1.0 in 1.7)
  - Service enhancements/deprecation overtime:
    - Probably need "compatibility/deprecation plan" for each service API (client API and wsdl)
      - "stable" vs "volatile"
      - "stable" shouldn't change between point releases (but can be deprecated)
        - Deprecated operations could be removed in 2 point releases
      - "volatile" can change between point releases
  - **Action Items:** We need to articulate the issues and proposed solutions in a presentation and give to the user community

- **Future Release Planning**
  - September/November Timeframe? 1.3
    - GME enhancements
    - Portal
      - Showing user what authorization they have to services
        - Ability to annotate services
        - Ability to know security requirements (not possible without a common authorization policy; trying to get buy in SWG)
      - Scalability
        - Adding DCQL builder (may add workflow support), how to deal with large data or large number of processes
          - **Portal Feature Request (fqp):** enumeration support for FQP
            - https://gforge.nci.nih.gov/tracker/index.php?func=detail&aid=4512&group_id=25&atid=177
      - **Portal Feature Request (portal):** user oriented operations to portal (change dorian password, request access to a grid grouper group)
    - **Portal Feature Request (gridgrouper):** need a way for a user to be able to request access to join a grid grouper group (currently may or may not be able to learn admin identities, but how can I ask the admin to join (can't learn email; would rather request it programmatically))
      - https://gforge.nci.nih.gov/tracker/index.php?func=detail&aid=13835&group_id=25&atid=177
    - Need a way to publish signed assertions
      - Coupled with trust network, can be used as a way for:
        - Users to publish information about themselves (name, institutions, etc), as an opt-in mechanism to addressing privacy issues (need to be careful about lack of identity vetted (eg LOA1) credentials being used to make assertions about physical identity-based attributes (e.g first/last name)
        - Services to publish information about themselves
      - **Action Item:** Flesh out use cases
        - **Scott**: metadata oriented
        - **Steve**: authorization oriented
    - Workflow notification and delegation, scufl execution integration
    - Ivy dealing with Globus location
    - Introduce
  - writeable APIs
  - HI7 datatypes
  - Security/CSM related
    - Dynamic policy, such as for writable APIs
    - Instance level support
    - Authz update/deprecate?
- caDSR team
  - GME Namespaces – there is still a lot to work out relative to validation, timing of the load, backout, exception handling, etc.
  - HL7 datatype implementation considerations
  - Semantic metadata registry futures discussion impact on the Grid – terminology metadata, services metadata, etc
- caGrid Transfer Integration session

Pasted from <file:///D:\projects\caBIG\caGrid-1-0\Documentation\management\osu\caGrid-Dev-F2F-Agenda-April2008.docx>

        - Deployment time security/policy extensions
        - Feasibility analysis on migration to GT4.2 and/or Axis2
        - Deploy time validation and extension (metadata enforcement, etc)
    - Transfer features per adoption, binary data format metadata
    - IDP discovery
    - Attribute Directory
    - Authz, CSM instance-level, etc
        - Break out authz 3 components (general PDP, pdp to map to CSM PEs, CSM group authz) to three projects. Deprecate CSM aspects (deal with new 4.x features separately), and make authz more "introduce friendly"
    - Data Service
        - Data Service with transfer
        - FQP with enumeration, transfer?, notification
        - Organizing data service projects
        - CQL2?

- **Authentication Service Discovery**
    - Need a way to support finding trusted authentication services (which Dorian would trust an assertion from) and which one a particular user should log into
        - The production Dorian IDP naming policy solves the second problem, in that the policy states the name needs to be descriptive to who is running it (e.g. Ohio State University)
        - For the first problem:
            - Could have Dorian have an operation to return the list of authentication services it trusts
            - Could have an assertion store service the information and dorian could issue the assertions to the store
                - Can we come up with a general way for assertions to be made and communicated (operation provider?) such that Dorian itself could issue them or an external service could issue them

- **caTissue CSM/Dorian account migration**
    - Presented overview and suggested minor adjustments
    - Detailed session on migration plugin process and roll out

- **Installer**
    - Headless Automated Installation
    - Front page "what do you want to do?"
        - Develop a grid service
        - Setup a container
        - Deploy Grid services
    - Should we get feedback from user community or domain workspaces
    - Help icons and text

- **CSM Authz**

  Action Items: **Vijay/Kunal**: review existing Authz documentation, provide writeup for what needs to be replaced to move to CSM 4x
  Action Items: **Shannon**: detail what we do for Resource authz, and how it could leverage CSM

- **Notes:**
    - CCTS is running 5 secure globus webapps in the same container without issues
    - We need to provide capability and/or process for monitoring availability and validation, and System's team need to take ownership of checking and responding to downtime or incorrect functionality
    - Any investigation or issues around things like denial of service? Do there need to be recommendations (e.g. not standing up data service against important transactional databases without ability to control/secure against DOS attacks)
        - Can we recommend a tool for testing a service (such as appscan that system's team uses)?
    - Need a better answer for "How do I get on the grid?"
        - Installer needs to be very good (e.g. MySQL's installer) and work for both experts (need to do lots of configuration such as for production grid) and for entry level (little to no configuration so they can get it up and running)
            - Do we need two different tools for this?
    - The was a discussion about how we could encrypt "data" for purposes of offline storage/delivery/and usage (essentially DRM)

- **Open Items:**
    - Discussion of how we can do smaller releases (redeploy prod grid?, upgrade all to new introduce?, etc)
        - Can we stage by releasing code and deploying to training/qa grid, let community try for a few months, then if needed make a point release and repeat, then finally move to production?